VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

MODELOVÁNÍ DYNAMIKY PROSODIE PRO ROZPOZNÁVÁNÍ ŘEČNÍKA

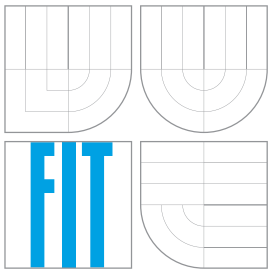
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. ZDENĚK JANČÍK

BRNO 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

MODELOVÁNÍ DYNAMIKY PROSODIE PRO ROZPOZNÁVÁNÍ ŘEČNÍKA

MODELLING PROSODIC DYNAMICS FOR SPEAKER RECOGNITION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. ZDENĚK JANČÍK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PAVEL MATĚJKA

BRNO 2008

Abstrakt

V současných systémech pro rozpoznání mluvčího se zpravidla využívají krátkodobé akustické příznaky. Jiné příznaky se používají jen zřídka. V práci se zaměřím na prosodické příznaky získané z průběhu základního tónu a energie. Tyto příznaky modelují průběh základního tónu v jednotlivých fonémech nebo slabikách. Z literatury je známo, že systémy založené na prosodii neposkytují tak dobré výsledky jako akustické, ale spojením akustického systému a systému založeného na prosodii se dosáhne značného zlepšení výsledků. To ověřím spojením s akustickým systémem vyvinutým na VUT. Při experimentech použiji data z evaluací pořádaných Národním úřadem pro standardy a technologie (NIST).

Klíčová slova

prosodie,základní tón,energie,identifikace mluvčího,ověření mluvčího, rozpoznání mluvčího, jazykový model, bigram, n-gram

Abstract

Most current automatic speaker recognition system extract speaker-depend features by looking at short-term spectral information. This approach ignores long-term information. I explored approach that use the fundamental frequency and energy trajectories for each speaker. This approach models prosody dynamics on single fonemes or syllables. It is known from literature that prosodic systems do not work as well the acoustic one but it improve the system when fusing. I verified this assumption by fusing my results with state of the art acoustic system from BUT. Data from standard evaluation campaigns organized by National Institute of Standarts and Technology are used for all experiments.

Keywords

prosody, pitch, energy, speaker identification, speaker validation, speaker recognition, language model, bigram, n-gram

Citace

Zdeněk Jančík: Modelování dynamiky prosodie pro rozpoznávání řečníka, diplomová práce, Brno, FIT VUT v Brně, 2008

Modelování dynamiky prosodie pro rozpoznávání řečníka

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením ing. Pavla Matějky.

.....
Zdeněk Jančík
12. května 2008

Obsah

Kapitola 1

Úvod

1.1 Motivace

První velkou skupinou uživatelů systémů pro rozpoznání řečníka jsou různé bezpečnostní složky (policie, armáda, apod.). Těm se hodí nahrávat, případně překládat odposlechy pouze, když mluví sledovaná osoba. Žádanou úlohou je také ověřování totožnosti osoby podle záznamu jejího hlasu.

Další využití se nabízí při prohledávání záznamů z porad, přednášek, schůzí a televizních pořadů, kde nás často zajímá nejen to co se říká, ale i kdo to říká. Můžeme vyhledávat pouze v promluvách dané osoby.

Komerčně zajímavým využitím je personalizace služeb. Kdy můžeme měnit přístup k zákazníkovi podle toho zda je to pravidelný zákazník nebo neznámá osoba.

1.2 NIST evaluace

Americká organizace NIST (National Institute of Standarts and Technology) pořádá od roku 1996 evaluace zaměřené na rozpoznání mluvčího. Těchto evaluací se účastní asi 40 organizací z celého světa a to jak z akademické sféry, tak z průmyslu. Je možné i spojení několika výzkumných týmů do jednoho celku.

Cílem je posunout výzkum v dané oblasti dále. Po každém ročníku následuje workshop, kde jednotlivý účastníci prezentují metody, jakými dosáhli svých výsledků. Tímto je zajištěn rychlý pokrok, protože každý účastník evaluace musí každý rok přijít s nějakým vylepšením aby uspěl.

Průběh evaluací rozpoznání mluvčího je popsán v [13]. Protože při vývoji systémů pro zpracování řeči je zvykem prezentovat výsledky na NIST datech, budu se i já v práci držet tohoto pravidla.

Kromě rozpoznání mluvčího NIST pořádá i evaluace zaměřené na detekci klíčových slov, rozpoznání jazyka a další (i mimo oblast zpracování řeči).

1.3 Stručný obsah práce

V kapitole 2 je definována úloha rozpoznání řečníka. Dále je uveden stručný popis nejpoužívanějších metod, aby bylo možno zasadit metody založené na prosodických příznacích do širšího kontextu. Je zde také popsán způsob jak hodnotit úspěšnost jednotlivých systémů.

V další kapitole (3) je vysvětlen pojem prosodie a popsán výběr a metody získání prosodických příznaků (základní tón, energie) pro rozpoznání mluvčího.

V kapitole 4 je podrobně rozebrána segmentace řeči do menších částí (od sebe odlišných slabik nebo fonémů) na základě průběhu základního tónu a energie, která bude použita pro rozpoznání mluvčího v systémech popsaných v kapitole 7.

Segmentaci je také možno získat z fonémového rozpoznávače, který přepisuje řeč na jednotlivé fonémy. Rozpoznání mluvčího založené na této segmentaci je v kapitole 8.

Při experimentech se systémem pro ropoznání mluvčího popsaných v kapitole 6 jsou použity pro modelování jazykové modely popsáné v kapitole 5.

Systémy pracující s prosodickými příznaky se používají hlavně ve spojení s ostatními systémy. Tímto spojením se zabývá kapitola 9.

V závěrečné kapitole 10 jsou vyhodnoceny výsledky experimentů, načrtnuto další pokračování projektu a naznačena souvislost s dalšímy projekty řešenými v rámci fakulty.

1.4 Návaznost na ročníkový projekt

V rámci ročníkového projektu jsem řešil segmentaci podle průběhu základního tónu popsanou v [4]. Při této příležitosti vznikla kapitola o rozpoznání řečníka (kap. 2), o prosodických příznacích (kap. 3) a kapitola zabývající se segmentací podle průběhu základního tónu (kap. 4).

Segmentace řešená v rámci ročníkového projektu je použita pro sestavení systémů využívajících segmentaci podle základního tónu (kap. 7).

Kapitola 2

Rozpoznání řečníka

Pro pojem rozpoznání mluvčího se často používají anglické zkratky SpeakerID nebo SpkID.

2.1 Základní úlohy SpkID

Podle [7, str. 489] můžeme ve SpkID identifikovat dvě základní úlohy verifikace řečníka a identifikace řečníka.

Při *verifikaci řečníka* ověřujeme zda daný hlas odpovídá danému řečníkovi. Vstupem systému je řečový signál a informace o identitě řečníka. Porovnáváme reprezentaci hlasu neznámého řečníka s reprezentací hlasu řečníka, za kterého se neznámý řečník vydává. Pokud se tyto reprezentace liší méně, než udává předem zadaný práh, prohlásíme o neznámém uživateli, že je uživatel jehož identitu zadal (z angl. client nebo target). V opačném případě uživatele prohlásíme podvodníkem (z angl. impostor nebo non-target).

V druhé úloze - *identifikaci řečníka* určujeme, který známý hlas nejlépe odpovídá hlasu neznámého řečníka. Taková úloha se pak nazývá identifikace na *uzavřené množině*. Pokud připustíme i případ, že neznámý řečník nemusí být žádným ze známých řečníků, jde o identifikaci na *otevřené množině*. Porovnání probíhá tak, že se vytvoří reprezentace hlasu neznámého řečníka a ta je porovnávána s reprezentacemi hlasů známých řečníků.

Pokud pracujeme na uzavřené množině, prohlásíme neznámého řečníka řečníkem s nejpodobnější reprezentací hlasu.

Pokud pracujeme na otevřené množině, porovnáme rozdíl reprezentací nejpodobnějšího hlasu a hlasu neznámého řečníka se zadaným prahem. Pokud je rozdíl větší než práh neodpovídá neznámý řečník žádnému ze známých řečníků.

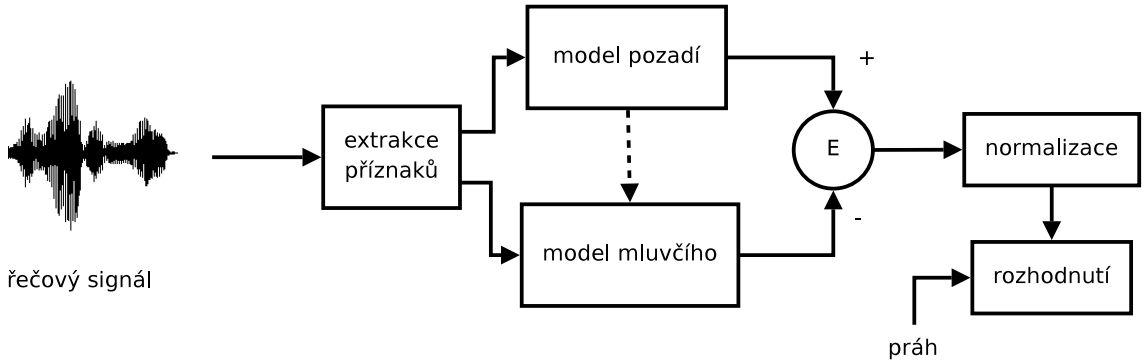
Úloha identifikace řečníka je rozšířením úlohy verifikace řečníka a proto se v praxi tyto úlohy často nerozlišují. Jejich striktní rozlišování je nutné až v nejvyšší úrovni abstrakce. Na nižších úrovních abstrakce jako je tvorba příznaků, jejich modelování apod. se postupuje u obou úloh stejně.

2.2 Dělení systémů pro SpkID

K SpkID existují dva přístupy: *textově závislé* a *textově nezávislé* rozpoznání. Při textově závislém rozpoznání musí uživatel vyslovit přesně zadanou promluvu. Při textově nezávislém rozpoznávání nezáleží na vyslovené promluvě. Tím, že předem známe promluvu dosahuje rozpoznání vyšší úspěšnosti, ale je pro některé úlohy nepoužitelné. Jde o úlohy, kdy řečník neví o SpkID systému, nebo záměrně nechce spolupracovat.

2.3 Schéma systému pro SpkID

Na obrázku 2.1 je vidět blokové schéma typického systému.



Obrázek 2.1: Blokové schéma systému pro SpkID

Vstupem je řečový signál. Ten je často *filtrován* a *segmentován*. Filtrací se upravuje spektrum signálu, případně se může odstraňovat šum apod. Segmentací se myslí dělení signálu na menší úseky (často se překrývající), ze kterých se extrahují *příznaky*, příznakový vektor. Ty mohou být různé (MFCC, LPC, základní tón ...). Typ použitých příznaků záleží na konkrétním přístupu k SpkID (podrobnosti jsou v kapitole 2.4).

Příznakový vektor je vyhodnocen společným modelem pro všechny mluvčí a modelem pro cílového mluvčího [1]. Abychom odstínili vliv prostředí při pořízení nahrávky počítáme rozdíl odezvy *modelu pro cílového mluvčího* a *modelu pro všechny mluvčí*[2]. Model pro všechny mluvčí se také s výhodou využívá při trénování. Tento model natrénujeme na datech pro všechny mluvčí, a modely pro cílové mluvčí trénujeme pouze adaptací modelu pro všechny mluvčí na daného mluvčího. Tento postup je nutný, protože téměř nikdy nemáme dostatek trénovacích dat od jednoho mluvčího na natrénování modelu cílového mluvčího.

Často se hodí mít věrohodnost (v angl. likelihood) pro danou promluvu v nějakém rozsahu, nebo je nutné namapovat výstupy jednotlivých modelů do jednoho rozsahu. Abychom tohoto dosáhli, *normalizujemee*. K tomu slouží například T-normalizace nebo Z-normalizace. Například T-normalizace je vypočtena jako

$$sc_{norm} = \frac{sc - mean}{\sqrt{var}} \tag{2.1}$$

kde sc je věrohodnost vrácena daným modelem pro danou promluvu. Hodnoty $mean$ a var jsou střední hodnota a rozptyl věrohodnosti promluvy testované proti ostatním modelům.

Normalizovanou likelihood *porovnáme s prahem* a podle výsledku porovnání přiřadíme dané promluvě (příznakovému vektoru) mluvčího, případně rozhodneme zda daná promluva pochází od daného mluvčího.

V praxi se pro zvýšení úspěšnosti často spojují výsledky více systémů založených na různých příznacích.

[1] Pokud provádíme identifikaci mluvčího neznáme model cílového mluvčího. Zkoušíme tedy postupně všechny modely a vybereme ten, který pro příznakový vektor vrací největší odezvu.

[2] v angličtině se používá termín Universal Background Model - UBM. V české literatuře se můžeme také setkat s označením model světa.

2.4 Možnosti jak dělat SpkID

2.4.1 Akustický přístup

Pro systém platí blokové schéma uvedené na obrázku 2.1. Jednotlivé systémy se liší v použitých příznacích a modelovacím přístupu. Příkladem systému používajícího akustický přístup může být systém BUT-GMM popsaný v kapitole 9.

Nejpoužívanější přístupy pro modelování jsou GMM (Gaussians mixture model [7, str. 521]), HMM (Hidden Markov model [7, str. 523]), neuronové sítě [10] a SVM (Support vector machine) [2].

Příznaky je možno dělit na nízkoúrovňové a vysokoúrovňové. *Nízkoúrovňové* příznaky se získávají jednodušeji a proto se začaly používat dříve. Jde o relativně jednoduše získatelné hodnoty zpravidla popisující krátký úsek řečového signálu. Mohou vycházet jak ze spektra signálu, tak z časové oblasti. Podle [7, str. 505] se nejprve používaly koeficienty lineární predikce (LPC). Později se začaly používat kepstrální koeficienty LPC a melovské kepstrální koeficienty (MFCC). Dnes jsou nejpoužívanější MFCC [7, str. 162].

2.4.2 Vysokoúrovňové SpkID

Do této skupiny patří systémy používající komplexnější příznaky, které se více blíží způsobu jakým rozpoznává jednotlivé mluvčí člověk. Jejich získání bývá složitější, proto se začaly pro rozpoznání mluvčího používat později. Často popisují delší úsek řečového signálu než nízkoúrovňové příznaky. Tyto příznaky se ukazují být komplementem k příznaků akustickým. Spojením výsledků systému založeného na akustickém přístupu s výsledky systému založeného na vysokoúrovňových příznacích významně zvýšíme úspěšnost rozpoznání mluvčího [4]. V [5] jsou vysokoúrovňové příznaky děleny na fonotaktické a prosodické.

Ve *fonotaktickém přístupu* se využívá výstup fonémového rozpoznávače pro trénování modelů pro cílového mluvčího a modelu světa. Modelujeme frekvence výskytů fonému (nebo sekvencí fonémů) v řeči.

Prosodickým příznakům se budu věnovat v následujích kapitole 3 a v kapitole 4, kde je více podrobností o jejich získávání a zpracování.

2.5 Hodnocení úspěšnosti systémů pro SpkID

Při rozpoznávání mluvčího mohou nastat čtyři možné výsledky:

- identita řečníka a zadaná identita se shodují a systém rozpozná řečníka dané identity, jde o *správné přijetí*
- identita řečníka a zadaná identita se shodují a systém nerozpozná řečníka dané identity, jde o *nesprávné odmítnutí*
- identita řečníka a zadaná identita se neshodují a systém rozpozná řečníka dané identity, jde o *nesprávné přijetí*
- identita řečníka a zadaná identita se neshodují a systém nerozpozná řečníka dané identity, jde o *správné odmítnutí*

Existují tedy dvě skupiny chyb, *chyby nesprávného odmítnutí* a *chyby nesprávného přijetí*. Potom můžeme systém charakterizovat pomocí poměru chyb nesprávného přijetí FAR

$$\mathrm{FAR} = \frac{n_{\mathrm{FA}}}{n_{\mathrm{impostor}}} \tag{2.2}$$

kde n_{FA} udává počet pokusů, kdy systém nesprávného řečníka přijal, a n_{impostor} je celkový počet pokusů, kdy měl systém měl odmítnout (šlo o podvodníka), a pomocí poměru počtu chyb nesprávného odmítnutí FRR

$$\text{FRR} = \frac{n_{\text{FR}}}{n_{\text{client}}} \tag{2.3}$$

kde n_{FR} udává počet pokusů, kdy systém správného řečníka odmítl, a n_{client} je celkový počet pokusů, kdy měl systém přijmout (šlo o správného řečníka).

V praxi se pro rychlé ohodnocení systémů hodí mít pouze jedno číslo. Tím je často míra rovnosti chyb *ERR* (Equal eRror Rate) daná vztahem 2.4, nebo hodnota ztrátové funkce *DCF* (Detection Cost Function) daná vztahem 2.6.

Pro určení míry rovnosti chyb hledáme takové nastavení prahu, kdy se míra chyb nesprávného odmítnutí a nesprávného přijetí rovná. Protože přesné nalezení hodnoty prahu Θ_{ERR} je obtížné, používá se přibližná hodnota ERR určená jako

$$\text{ERR} = \frac{\text{FRR}(\Theta'_{\text{ERR}}) + \text{FAR}(\Theta'_{\text{ERR}})}{2} \tag{2.4}$$

kde Θ'_{ERR} určíme jako

$$\Theta'_{\text{ERR}} = \text{argmin}|\text{FRR}(\Theta) - \text{FAR}(\Theta)| \tag{2.5}$$

Hodnotu ztrátové funkce (DCF) pro daný práh určíme jako

$$\text{DCF} = \text{cost}(\text{FR}) \cdot \text{FRR} \cdot P(\text{client}) + \text{cost}(\text{FA}) \cdot \text{FAR} \cdot P(\text{impostor}) \tag{2.6}$$

kde cost(FR) značí cenu nesprávného odmítnutí a cost(FA) cenu nesprávného přijetí. Hodnota $P(\text{client})$ označuje apriorní pravděpodobnost, že jde o správného řečníka a $P(\text{impostor})$ je apriorní pravěpodobnost, že jde o podvodníka a zároveň platí

$$P(\text{impostor}) = 1 - P(\text{client}) \tag{2.7}$$

Při hodnocení systémů pro SpkID je kromě úspěšnosti nutné udat za jakých podmínek bylo těchto výsledků dosaženo. Jde o vlastnosti soustavy pro nahrávání dat (mikrofony, vzorkovací frekvence, podmínky při nahrávání apod.), množství trénovacích dat pro natrénování modelu, nastavení prahu (obr. 2.1) atd.

2.5.1 Grafická podoba hodnocení

Číselné hodnoty FRR a FAR jsou závislé na nastavení prahu Θ. Pro grafické porovnání výsledků systémů pro SpkID se často používá *ROC křivka* (Receiver Operating Characteristic). Ta zobrazuje závislost FRR na FAR, přičemž hodnota Θ je parametrem závislosti.

Pokud do jednoho obrázku zaneseme více ROC křivek, s malými rozdíly měr FRR a FAR budou tyto křivky příliš blízko u sebe a obrázek se stane nepřehledným. Tento jev je ještě více posílen pro malé hodnoty měr FRR a FAR.

Řešením je použití křivky DET (Detection error Trade-off curve), zavedené v [6], která má na osách kvantily normovaného normálního rozdělení, které odpovídají poměrným počtům chyb FRR a FAR. Tímto postupem dostáváme nelineární osy.

Úspěšnost systému je tím lepší čím více se průběh ROC nebo DET křivky blíží k počátku.

Kapitola 3

Prosodie

V [7, str. 64] se k prosodii píše:

> Termínem *prosodie* se označují takové vlastnosti řečového signálů, které souvisí především s *frekvencí základního tónu* (výškou hlasu), *intenzitou* (hlasitostí) a *časováním*. Mezi aspekty řeči spojené s časováním patří rytmus (a s tím spojené rozvržení přízvuku) a rychlost řeči (trvání slabik a hlásek). Změny základního hlasivkového tónu tvoří melodii (resp. intonaci) promluvy. Prosodické rysy se přitom vztahují k větším řečovým jednotkám (jako jsou slabiky, slova, celé věty, nebo dokonce i větší promluvy).

3.1 Základní tón

Frekvence základního hlasivkového tonu odpovídá frekvenci prvního formantu (nabývá hodnot asi od 50Hz u mužů až po 400Hz u dětí). Základní tón se udává buď v hertzích. Často označuje jako f_0 [1]. Můžeme se setkat i s pojmem perioda základního tónu, která se udává ve vzorcích nebo sekundách. O základním tónu má smysl mluvit pouze u znělých hlásek.

Naivní jednoduchá metoda filtrovat signál dolní propustí nelze použít, díky značnému rozsahu hodnot základního tónu. Pokud bychom zvolili lomovou frekvenci například 100 Hz, nedetekovali bychom vyšší hodnoty a pokud bychom zvolili vyšší hodnotu lomové frekvence, tak by se ve spektru objevovali i vyšší harmonické.

Pro detekci základního tónu se využívá poznatek, že jde o první maximum výkonového spektra. Základní tón se určuje vždy pro jeden znělý rámec[2].

3.1.1 Autokorelační metoda

Pro řečový rámec spočítáme autokorelační funkci

$$R(m) = \sum_{n=0}^{N-1-m} s(n)s(n+m) \tag{3.1}$$

Základní tón je vzdálenost maxima R_{max} od nultého koeficientu $R(0)$. Pokud je hodnota maxima R_{max} výrazně menší než hodnota $R(0)$ je rámec neznělý.

[1] V anglické literatuře se můžeme setkat s pojmem *pitch*.

[2] v angličtině označovaný jako *frame*, používá se délka 20 až 30 ms a přesah 10 ms pro vzorkovací frekvenci 8 kHz.

Tato metoda často nenalezne správnou hodnotu. V praxi se nepoužívá. Příklad průběhu základního tónu vygenerovaného autokorelační metodou je v prvním grafu na obrázku 3.1.

3.1.2 Kroskorelační metoda

Odstraňuje problém se zmenšujícím se počtem vzorků, ze kterých jsou počítány vyšší autokorelační koeficienty zahrnutím předchozího rámce do výpočtu.

$$CCF(m) = \sum_{n=zr}^{zr+N-1} s(n)s(n-m) \tag{3.2}$$

Tím se ovšem změní celková energie signálu. To můžeme řešit normalizací

$$NCCF(m) = \frac{\sum_{n=zr}^{zr+N-1} s(n)s(n-m)}{\sqrt{E_1 E_2}} \tag{3.3}$$

kde

$$E_1 = \sum_{n=zr}^{zr+N-1} s^2(n) \qquad E_2 = \sum_{n=zr}^{zr+N-1} s^2(n-m) \tag{3.4}$$

Metoda poskytuje lepší výsledky než autokorelace, ale výsledky jsou stále v praxi nepoužitelné. Výstupem jsou často dvojnásobné nebo vícenásobné hodnoty vlivem nedostatečného potlačení vlivu vyšších formantů.

Příklad průběhu základního tónu vygenerovaného normalizovanou kroskorelační metodou je na obrázku 3.1 v prvním grafu.

3.1.3 Program wavesurfer

Ve zpracování řeči často používaný program pro rychlý náhled na řečový signál. Má více funkcí než jen detekci základního tónu.

Pro detekci základního tónu požívá knihovnu ESPS (Entropic Speech Processing System) kde je implementována metoda AMDF (Average Magnitude Difference Function).

Program šířen ve jako open source, oficiální stránky jsou na adrese `http://www.speech.kth.se/wavesurfer/`. Program se již nevyvíjí. Nevýhodou wavesurferu je jeho nestabilita při zpracování delších nahrávek než několik desítek sekund.

Příklad průběhu základního tónu vygenerovaného programem wavesurfer je na obrázku 3.1 v druhém grafu.

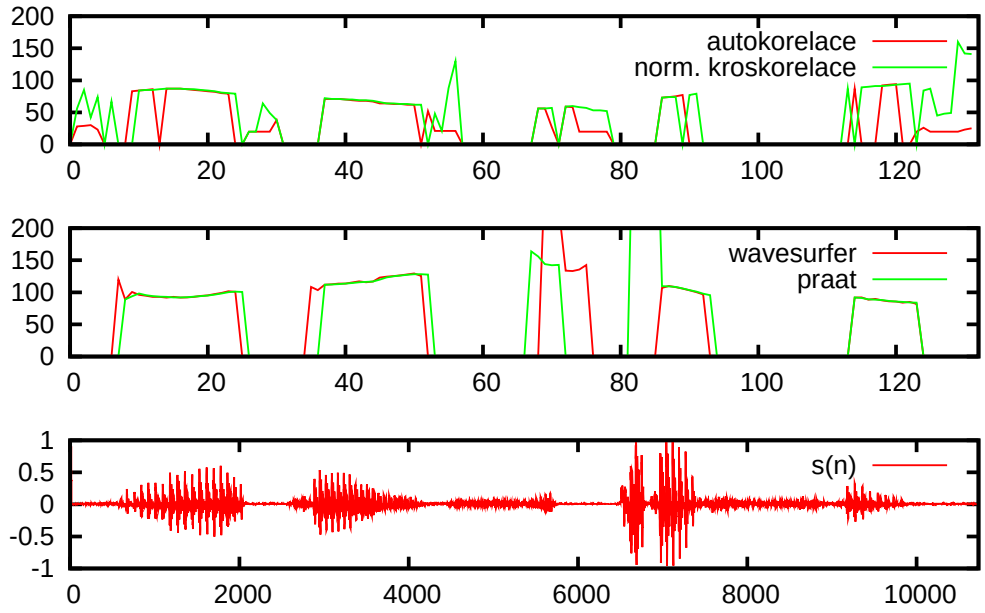
3.1.4 Program praat

Další z programů používaný ve zpracování řeči. Má ve srovnání s programem wavesurfer větší množství funkcí a má možnost spuštění z příkazové řádky. Je složitější na ovládání a detekce základního tónu mu trvá déle než wavesurferu. Základní výhody proti wavesurferu jsou stabilita a subjektivně lepší výsledky detekce základního tónu.

Detekce základního tónu v praatu je založena na přesné autokorelační metodě (algoritmus v dodatku A). Podrobný popis metody detekce základního tónu a srovnání s dalšími metodami je v článku [1].

Program je šířen pod licencí GNU GPL, oficiální stránky jsou na adrese `http://www.fon.hum.uva.nl/praat/`.

Příklad průběhu základního tónu vygenerovaného programem praat je na obrázku 3.1 v druhém grafu.



Obrázek 3.1: Srovnání metod detekce základního tónu

3.1.5 Srovnání detektorů f_0

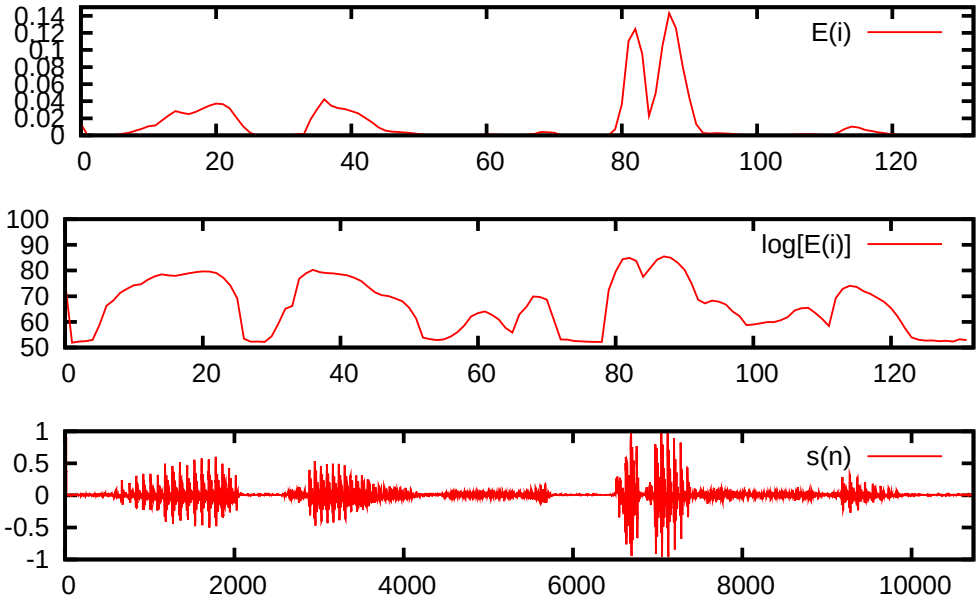
Detektory základního tónu je nutno porovnávat subjektivně, protože data označená základním tónem jsou velmi těžko dostupná. Požadavky na detektor použitelný pro SpkID:

- měl by produkovat vyhlazený průběh
- neměl by detekovat vyšší násobky f_0
- místa s malou amplitudou signálu by měl označit jako neznělá
- možnost ovládání z příkazové řádky
- vysoká rychlost detekce a malé nároky na paměť počítače

Na obrázku 3.1 jsou porovnány různé metody detekce základního tónu. V prvním grafu je porovnání autokorelační metody s metodou kroskorelační. V druhém grafu jsou porovnány programy wavesurfer a praat. V posledním grafu je vidět signál, na kterém bylo srovnání prováděno.

Z výsledků je vidět nepoužitelnost základních metod. Tyto metody jsou ovšem základem pro tvorbu pokročilejších metod.

Programy praat a wavesurfer poskytují srovnatelné výsledky, ale program praat umožňuje ovládání z příkazové řádky a je stabilnější. Proto byl pro další experimenty vybrán program praat.



Obrázek 3.2: Krátkodobá energie signálu a její logaritmovaná podoba

3.2 Krátkodobá energie

Kolísání hlasitosti můžeme modelovat průběhem krátkodobé energie. Ta je pro i-tý rámec definována jako:

$$E(i) = \sum_{j=0}^{N} s^2(iN + j) \tag{3.5}$$

kde N je délka řečového rámce ve vzorcích a $s(n)$ je řečový signál.

V obrázku 3.2 je průběh energie počítané podle výše uvedeného vztahu na prvním grafu. Z důvodu značného dynamického rozsahu hodnot a vlastností lidského sluchu je lepší pracovat s logaritmickou energií podle vztahu:

$$E_{log}(i) = \log\left(\sum_{j=0}^{N} s^2(iN + j)\right) \tag{3.6}$$

Průběh E_{log} je na obrázku 3.2 uveden jako druhý graf. Třetí průběh je řečový signál, pro který je počítána energie.

Kapitola 4

Segmentace podle f_0

Pro rozpoznání mluvčího je třeba získat příznakový vektor charakterizující daného mluvčího. K tomu vedou následující kroky:

1. Rozdělení řečového signálu na rámce. Pro každý rámec je určen základní tón (kap. 3.1) a energie.(kap. 3.2).
2. Průběh základního tónu je vyhlazen a rozdělen na segmenty (kap. 4.1).
3. Krátké segmenty jsou eliminovány (kap. 4.2).

4.1 Základní segmentace a vyhlazení

Abychom zabránili vzniku velkého množství segmentů a odstranili šum provádíme vyhlazení průběhu základního tónu. K tomu se dají využít dvě metody. *Průměrovací* filtr a *mediánový* filtr.

Průměrovací filtr

$$y(i) = \frac{1}{2k+1} \sum_{n=-k}^{k} x(i+k) \tag{4.1}$$

počítá aritmetický průměr vzorků v okénku délky $2k+1$.

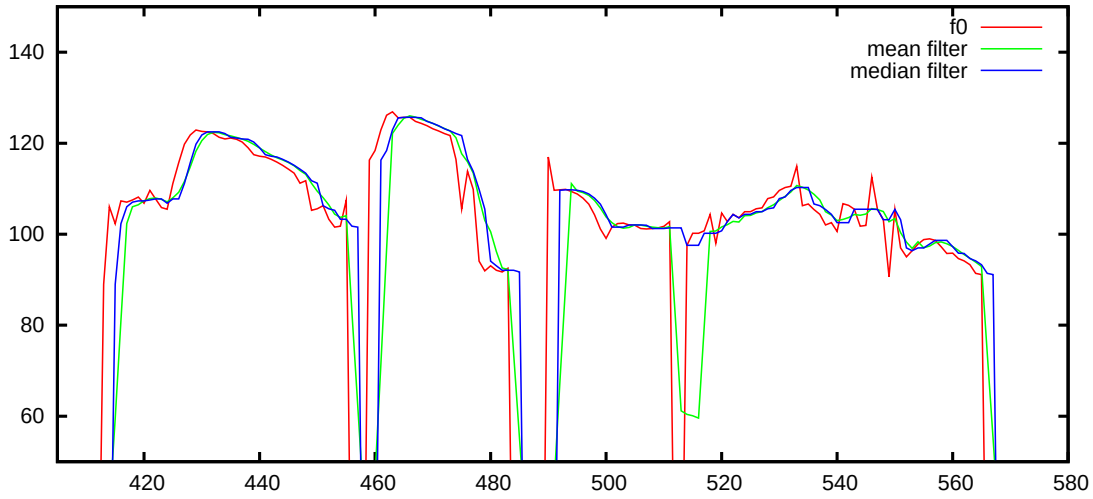
Mediánový filtr

$$y(i) = \text{median}(i-k : i+k) \tag{4.2}$$

počítá medián vzorků na okénku délky $2k+1$.

Z obrázku 4.1 je vidět, že se nejlepší výsledky poskytuje mediánový filtr, proto je použit v následujích experimentech. Průměrovací filtr nezachovává svislé hrany a nedostatečně vyhlazuje průběh signálu. Mediánový filtr tyto hrany posunuje o k vzorků, což ovšem lze jednoduše kompenzovat.

Pro každý vzorek filtrovaný mediánovým filtrem $y(i)$ spočítáme jeho odchylku od předchozího vzorku $\Delta y(i) = y(i-1) - y(i)$. Pokud je odchylka kladná označíme trend vzorku jako kladný, pokud je záporná, jako záporný a pokud je nulová označíme trend vzorku jako ticho. Hranici segmentu poznáme jako změnu trendu vzorků. Tímto postupem dostaneme signál dělený na segmenty se stejným trendem.



Obrázek 4.1: Srovnání mediánového a průměrovacího filtru

4.2 Spojování segmentů

Základní segmentací (kap. 4.1) vznikne značný počet krátkých segmentů[1]. Blokové schéma algoritmu pro eliminaci krátkých segmentů je na obrázku 4.2.

Nejprve načteme segment. Rozhodneme zda je následující segment krátký nebo dlouhý. Pokud jde o dlouhý segment, tak vypíšeme první načtený segment a pokračujeme v načítání dalšího segmentu.

Pokud jde o krátký segment pokračujeme v načítání krátkých segmentů, které spojujeme s dlouhým segmentem tak dlouho dokud nenajdeme další dlouhý segment.

Pokud je následující dlouhý segment stejný jako ten předchozí, oba segmenty spojíme a pokračujeme zjišťováním zda je další segment krátký. Pokud se segmenty liší spojíme první dlouhý segment s následujícími krátkými segmenty a vypíšeme takto získaný segment. S druhým dlouhým segmentem pokračujeme ve zjišťování, zda je další segment krátký.

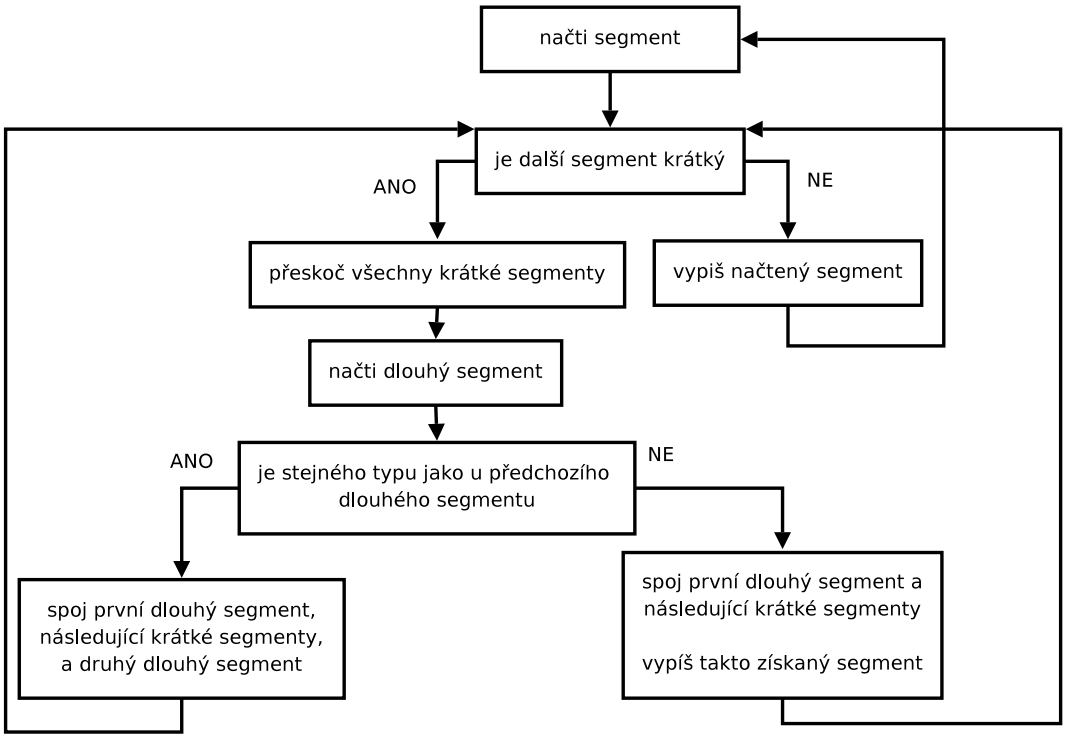
4.3 Tvorba příznaků

Na obrázku 4.3 je vidět dělení signálu na rámce podle trendu základního tónu v prvním grafu. Průběh energie (druhý graf) je dělen na stejné segmenty jako základní tón. Trend segmentu je určen jako průměrný trend. Spočítá se suma rozdílů dvou po sobě následujících hodnot energie a podle znaménka této sumy je určen trend energie stejně jako trend základního tónu v kap. 4.1.

Trend základního tónu a energie označíme jako:

- *(+)* pokud trend stoupá - odchylka dvou po sobě následujících vzorků je *kladná*
- *(-)* pokud trend klesá - odchylka dvou po sobě následujících vzorků je *záporná*
- *(uv)* pokud je trend nulový - odchylka dvou po sobě následujících vzorků je *nulová*

[1] Za krátký segment je považován segment dlouhý jeden rámec. Tato hodnota je ověřena experimenty popsanými v kapitole 7.1.



Obrázek 4.2: Algoritmus eliminace krátkých segmentů

Pokud je trend_f0 roven nule, jde o neznělý úsek řeči, nebo ticho. Takovým úsekům se nepřiřazuje žádné označení pro trend energie.

Takto pro každý segment vygenerujeme jedno z možných označení z množiny $\{++, +-, -+, --, uv\}$.

První segment na obrázku 4.3 má trend_f0 roven 0 proto je označen jako `uv` (je neznělý, a proto pro trend_E není generován příznak). Následující segment má trend základního tónu a energie kladný je tedy označen `++`. Pro příklad na obrázku 4.3 dostáváme tuto sekvenci příznaků:

```
uv ++ -- +- -- uv ++ -- ++ -- +- -- uv ++
```

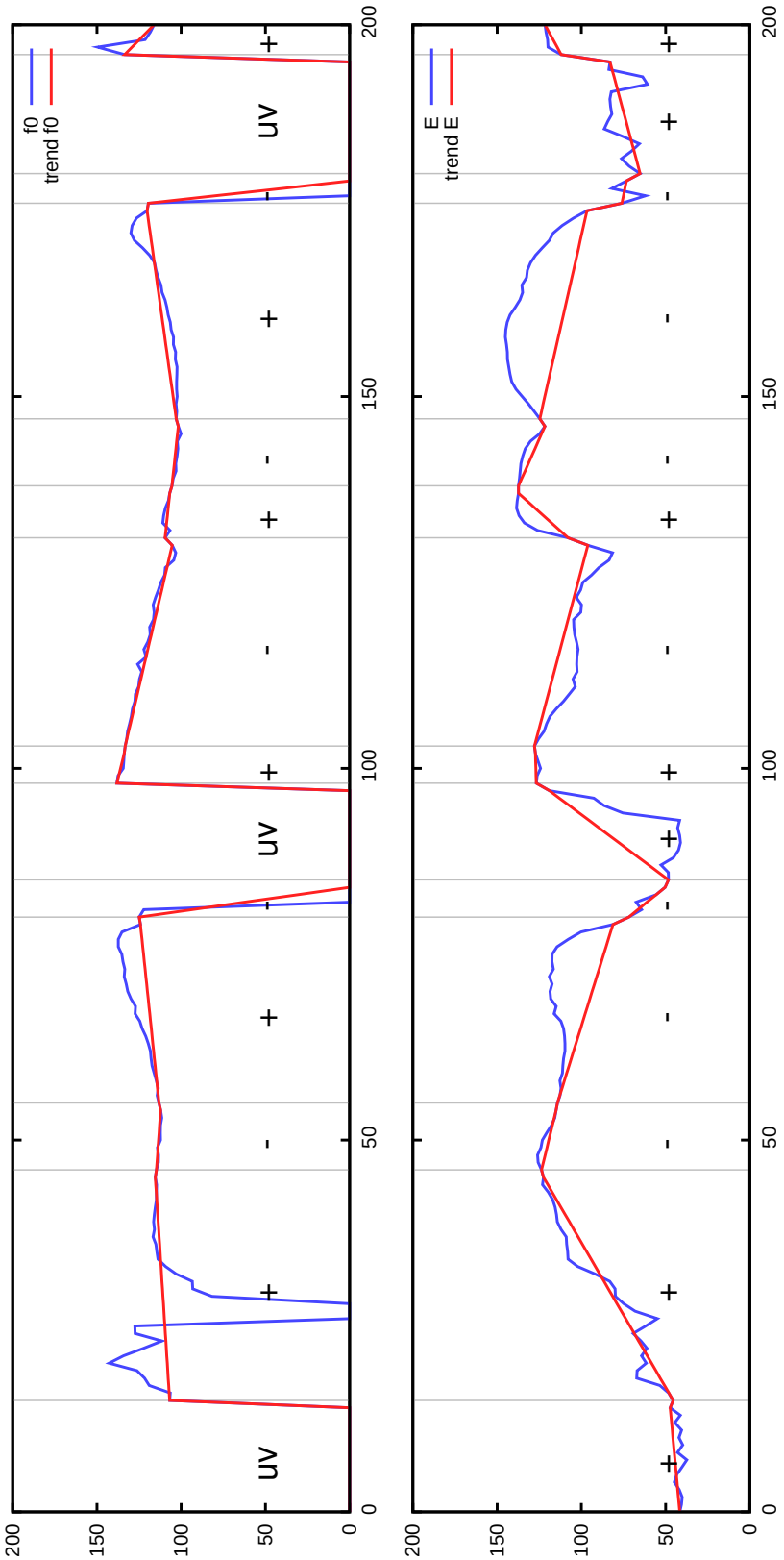
V prvním grafu na obrázku 4.3 je modrou barvou zobrazen průběh základního tónu (resp. energie) bez filtrace mediánovým filtrem. Červenou barvou je vykreslen průběh trendu základního tónu (resp. energie). Hranice segmentů je zobrazena šedou barvou. U každého segmentu je také zobrazen jeho trend.

Na začátku průběhu (kolem vzorku 30) je vidět spojení dvou segmentů. Ke spojení došlo, protože mezera mezi segmenty je kratší než minimální délka segmentu.

Na průběhu základního tónu přibližně mezi rámci 100 a 150 je zřetelně vidět vyhlazení průběhu a eliminace krátkých segmentů.

4.4 Délka segmentu

Samotná délka segmentu není pro rozpoznávání vhodná, protože segmenty různé délky jsou kvalifikovány jako odlišné i když se délka liší třeba jen o jeden nebo dva vzorky. Řešením je



Obrázek 4.3: Výsledek segmentace

kvantování délek segmentů do několika hladin. V článku [4] jsou použity tři hladiny (krátký, střední a dlouhý segment).

Pro určení kvantizačních úrovní je nutné vypočítat histogram délek segmentů v trénovacích datech. Na základě histogramu určíme kvantizační hranice tak, aby na každou z kvantizačních úrovní připadlo 33 % segmentů.

Například pro data z obrázku 4.3 dostáváme následující segmentaci:

```
uv(13) ++(29) --(7) +-(23) --(3) uv(11) ++(3)
--(26) ++(5)  --(7) +-(27) --(2) uv(14) ++(3)
```

kde v závorkách je délka segmentu. Pokud krátký segment označíme jako `S`, střední `M` a dlouhý `L`, tak pro hranice 4 a 8 vzorků dostaneme sekvenci příznaků:

```
uvL ++L --M +-L --S uvL ++S --L ++M --M +-L --S uvL ++S
```

Kapitola 5

Jazykový model

Úkolem jazykového modelu je určit pro každou posloupnost slov[1] W apriorní pravděpodobnost $P(W)$. Pravděpodobnost posloupnosti K slov můžeme určit jako

$$P(w_1 w_2 w_3 \dots w_k) = P(w_1)P(w_2|w_1)\dots P(w_K|w_1 \dots w_{K-1}) = \prod_{i=1}^{K} P(w_i|w_1^{i-1}) \tag{5.1}$$

Znalost celé historie by byla pro praktické nasazení modelu nepoužitelná, a proto je tato historie omezena na n posledních slov. Vycházíme z předpokladu, že omezení historie příliš nezhorší výsledky.

$$P(w_1 w_2 w_3 \dots w_k) \approx \prod_{i=1}^{K} P(w_i|w_{i-n+1} \dots w_{i-1}) \tag{5.2}$$

Takový model potom nazýváme n-gramovým jazykovým modelem. Pro $n = 1$ máme označení *unigram*, pro $n = 2$ *bigram* a pro $n = 3$ *trigram*. Nejčastěji používané jsou modely bigramové a trigramové.

5.1 Trénování

Pro odhad parametrů n-gramového modelu na trénovacím korpusu se používá vztah

$$P(w_i|w_{i-n+1} \dots w_{i-1}) = \frac{N(w_{i-n+1}, \dots, w_i)}{N(w_{i-n+1}, \dots, w_{i-1})} \tag{5.3}$$

kde $N(w_{i-n+1}, \dots, w_i)$ je počet výskytů n-gramu $w_{i-n+1}, \dots, w_i$ v trénovacích datech a $N(w_{i-n+1}, \dots, w_{i-1})$ je počet výskytů $(n-1)$-gramu $w_{i-n+1}, \dots, w_{i-1}$.

Ze vztahu 5.3 vyplývá nevýhoda n-gramových modelů. Nelze přímo určit pravděpodobnost n-gramu, pokud se tento n-gram v trénovacích datech ani jednou neobjevil.

To se řeší snížením pravděpodobností n-gramů, které se v trénovacích datech objevily a zvýšením pravděpodobností n-gramům, které se v trénovacích datech neobjevily. Tomuto postupu se říká *vyhlazování*. Vyhlazování jazykového modelu je otevřený problém. Základní metody vyhlazování jsou v [7, str.232].

[1] Slovem je zde myšlen jakýkoliv řetězec daného jazyka. Jazykem L označujeme podmnožinu $L \subseteq \Sigma^*$, kde množina Σ^* obsahuje všechny řetězce nad abecedou Σ. Slovo v tomto kontextu tedy nemusí odpovídat slovu přirozeného jazyka, ale může být například textovou reprezentací příznakového vektoru.

5.2 Příklad bigramového modelu

Pro trénovací sekvenci

```
uv -+ +- uv +- uv -+ uv +- uv +- uv -+ +- uv -+ +-
uv -+ +- uv ++ uv ++ -- uv ++ -- ++ uv ++ -+ uv -+
uv -+ uv ++ -- uv -+ ++ -+ +- uv ++ -+ uv +- uv -+
+- -- uv -+ +- -- uv -+ uv -+ uv
```

určíme pravděpodobnosti jednotlivých bigramů vyplněním tabulky 5.1. Bigramy seskupujeme v tabulce podle kontextu (první slovo bigramu). Pro každý bigram určíme jeho četnost v testovacích datech. Sečteme četnosti v rámci kontextu. Výsledná pravděpodobnost bigramu se určí (podle vztahu 5.3) jako četnost bigramu lomeno celková četnost přes celý kontext.

bigram	četnost	pravděpodobnost
`uv -+`	12	12/21
`uv +-`	3	3/21
`uv ++`	6	6/21
celkem	21	
`-+ uv`	6	6/14
`-+ +-`	7	7/14
`-+ ++`	1	1/14
celkem	14	
⋮	⋮	⋮

Tabulka 5.1: Určení pravděpodobností bigramů na trénovací promluvě

Pokud máme určit pravděpodobnost testovací promluvy:

```
uv +- uv -+ ++ -- +- uv -+ ++ -+ +- uv -- ++ uv -+
```

proti výše natrénovanému modelu vyplňujeme tabulku 5.2.

Pro každý bigram trénovací promluvy určíme jeho četnost, kterou vynásobíme pravděpodobností tohoto bigramu z tabulky 5.1. Celková pravděpodobnost promluvy je potom součinem pravděpodobností jednotlivých bigramů.

bigram	pravděpodobnost bigramu	
`uv +-`	$1 \cdot 3/21$	$= 1/7$
`uv -+`	$3 \cdot 12/21$	$= 12/7$
`uv --`	$1 \cdot 0/21$	$= 0$
`-+ ++`	$2 \cdot 1/14$	$= 1/7$
`-+ +-`	$1 \cdot 7/14$	$= 1/2$
⋮	⋮	⋮

Tabulka 5.2: Určení pravděpodobnosti testovací promluvy

V tabulce 5.1 není uvedena pravděpodobnost bigramu `uv --`, protože se neobjevil v trénovacích datech. Tento problém se řeší pomocí tzv. vyhlazování.

Tabulky 5.1 a 5.2 jsou pro zjednodušení zkráceny.

Kapitola 6

Data a programové nástroje

Každý provedený experiment bude obsahovat tyto kroky:

1. segmentace, sestavení příznaků pro trénovací i testovací sadu
2. natrénovaní jazykového modelu pro každého mluvčího a modelu UBM
3. testovaní modelů podle testovacího předpisu
4. vyhodnocení výsledků

6.1 Data

Jde o telefonní data namluvená dobrovolníky a nahraná se vzorkovací frekvencí 8 kHz v kódování μ-law. Každý soubor obsahuje záznam jednoho rozhovoru, kde v levém kanálu mluví první mluvčí a v pravém kanálu druhý mluvčí. Před spuštěním experimentů je nutné každý kanál uložit do jednoho souboru.

Pokud není uvedeno jinak jsou v experimentech používaná pro trénování a testování NIST SRE 2006 data. Pro srovnání byly některé experimenty provedeny i na NIST SRE 2005 datech.

6.1.1 NIST SRE 2005

Trénovací sada obsahuje data od 646 mluvčích, kde je 372 žen a 274 mužů. Pro každého mluvčího trénovací sada obsahuje jednu promluvu dlouhou přibližně dvě a půl minuty.

Testujeme verifikaci mluvčího na 31 194 testovacích párech model mluvčího, soubor, přičemž testovací sada obsahuje 2 751 testů kdy soubor patří danému mluvčímu a 27 951 kdy jde o jiného mluvčího.

6.1.2 NIST SRE 2006

Trénovací sada obsahuje data od 816 mluvčích, kde je 462 žen a 354 mužů. Pro každého mluvčího trénovací sada obsahuje jednu promluvu dlouhou přibližně dvě a půl minuty.

Testovací sada předepisuje 53 966 testovacích dvojic model mluvčího, soubor, přičemž testovací sada obsahuje 1 854 testů kdy soubor patří danému mluvčímu a 22 159 kdy jde o jiného mluvčího.

6.2 SRILM

Pro trénování a testování jazykových modelů jsem použil SRILM (The SRI Language Modeling Toolkit) popsaný v [9]. Jde o implementaci n-gramových modelů používající Good-Turingův odhad a Katzův diskontní model pro vyhlazování. Volbou parametrů je možno aplikovat i jiné vyhlazovací postupy.

6.2.1 Trénování

Pro trénování jazykového modelu se používá příkaz

```
ngram-count -text train.sri -order 2 -lm speaker.lm
```

kde soubor `train.sri` obsahuje pro každý trénovací soubor jeden řádek, na kterém jsou výsledky segmentace. Začátek jednoho z trénovacích souborů (`kcbz_a.seg`) má tvar:

```
uv -- +- -- ++ -- uv -+ +- -- +- uv -+ +- -- +- uv -+ ++
```

Parametr `order` udává řád n-gramového modelu. Natrénovaný model je uložen do souboru `speaker.lm`. Výše uvedený soubor `kcbz_a.seg` je použit pro trénování modelu mluvčího T1442, který potom má potom tvar:

```
\data\
ngram 1=7
ngram 2=18

\1-grams:
-0.8084025 ++  -1.374973
-0.7423826 +-  -1.427503
-0.7652456 -+  -1.481132
-0.638842  --  -1.606071
-2.634477  </s>
-99        <s> -0.3464347
-0.5852593 uv  -1.64391

\2-grams:
-0.9420081 ++ -+
-0.1208222 ++ --
-0.9420081 ++ uv
-1.198657  +- -+
-0.3294254 +- --
-0.3413246 +- uv
-0.5740313 -+ ++
-0.2730013 -+ +-
-0.7289332 -+ uv
-0.6777807 -- ++
-0.60206   -- +-
-0.2757241 -- uv
-0.1760913 <s> uv
-0.6381051 uv ++
-0.9391351 uv +-
-0.2537379 uv -+
-1.098836  uv --
-1.92814   uv </s>

\end\
```

6.2.2 Adaptace z UBM

Protože pro kompletní natrénování jazykového modelu pro každého mluvčího je nutné velké množtví dat, které nejsou k dipozici, bylo nutné použít adaptaci z UBM. Míru adaptace modelu ovlivňujeme hodnotou *lambda*, udávající váhu modelu cílového mluvčího.

V grafech s výsledky testů je *lambda* uvedena na ose x.

6.2.3 Testování

Pro testování modelu mluvčího se používá příkaz

```
ngram -order 2 -lm speaker.lm -mix-lm UBM.lm -lambda 0.5 -ppl speaker.tst
```

Parametr `order` udává řád n-gramového modelu použitého pro test. Testovaný jazykový model se načítá ze souboru `speaker.lm`. Parametry `-lambda` a `-mix-lm` se používají pro adaptaci modelu mluvčího z modelu UBM (viz. 6.2.2). Soubor `speaker.tst` obsahuje testovací data, kde výsledek segmentace pro každý testovaný soubor je na jednom řádku.

Segmentací dostaneme pro každý soubor jiný počet segmentů. Aby bylo možné jednotlivé výsledky porovnávat je je nutno normalizovat počtem segmentů.

Výsledkem testování je pro každý soubor logprob (celková logaritmická pravděpodobnost), ppl (perpelxita) a ppl1 (perplexita 1), ze které nás zajímá `ppl1`. Vyjadřující

$$\text{ppl1} = 10^{-\text{logprob}/\text{words}} \tag{6.1}$$

kde *logprob* je celková logaritmická pravděpodobnost modelu a *words* je počet segmentů v souboru.

Například výsledkem testování souboru `oamp_a.seg` proti modelu mluvčího T1442 následující výstup programu `ngram`:

```
file data/NIST-SRE/test/segment-data/oamp_a.seg: 1 sentences, 1039 words, 0 OOVs
0 zeroprobs, logprob= -466.169 ppl= 2.80698 ppl1= 2.80977
```

Z výstupu je vidět, že soubor `oamp_a.seg` obsahuje 1039 segmentů. Pro normalizaci délkou segmentu by bylo nutné dělit logprob touto hodnotou. Než z výstupu extrahovat počet segmentů a tím dělit logprob, je jednoduší extrahovat perplexitu ppl1 a tu logaritmovat.

Každý testovací soubor testujeme jak proti modelu mluvčího ($\text{ppl1}_{\text{speaker}}$) a proti UBM ($\text{ppl1}_{\text{ubm}}$), výsledné skóre je potom

$$score = \log(\text{ppl1}_{\text{ubm}}) - \log(\text{ppl1}_{\text{speaker}}) \tag{6.2}$$

Pokud je *score* kladné je soubor přijat (testovací soubor odpovídá danému modelu).

6.3 Evaluace výsledků

Výsledky se vyhodnocují skriptem používaným pro vyhodnocování výsledků NIST SRE evaluace 2005 a 2006. Tento skript porovná výsledky testování s referenčními a vypočítá hodnoty ERR (Equal eRror Rate) a DCF (Detection Cost Function) a vykreslí DET křivku (Detection error trade-off). Více podrobností o problému evaluace systémů pro rozpoznání řečníka je v kapitole 2.5.

Kapitola 7

Systémy založené na segmentaci podle f_0

V článku [4] jsou popsány postupy rozpoznání mluvčího založené na prosodických příznacích. V *základním systému* je použit příznakový vektor obsahující pouze dvojici trend základního tónu a trend energie. Tento systém byl dále rozšířen o *délku segmentu*.

Kromě segmentace podle základního tónu je možno segmentovat i *podle energie*. Tento přístup je také ověřen.

7.1 Základní systém f_0, E

7.1.1 Vyhlazování průběhů f_0 a E

Pro vyhlazování průběhu základního tónu je použit mediánový filtr. V průbězích základního tónu se objevují pauzy kratší než je polovina okna mediánového filtru. Tyto krátké segmenty ticha by pak byly potlačeny, proto je možno modifikovat mediánový filtr tak, aby se neaplikoval na vzorky označující ticho.

7.1.2 Ladění parametrů segmentace

Výslednou segmentaci ovlivňuje nastavení konstanty *minseg*, určující minimální délku segmentu, který je v algoritmu spojování segmentů (je popsán v kap. 4.2) označen jako krátký.

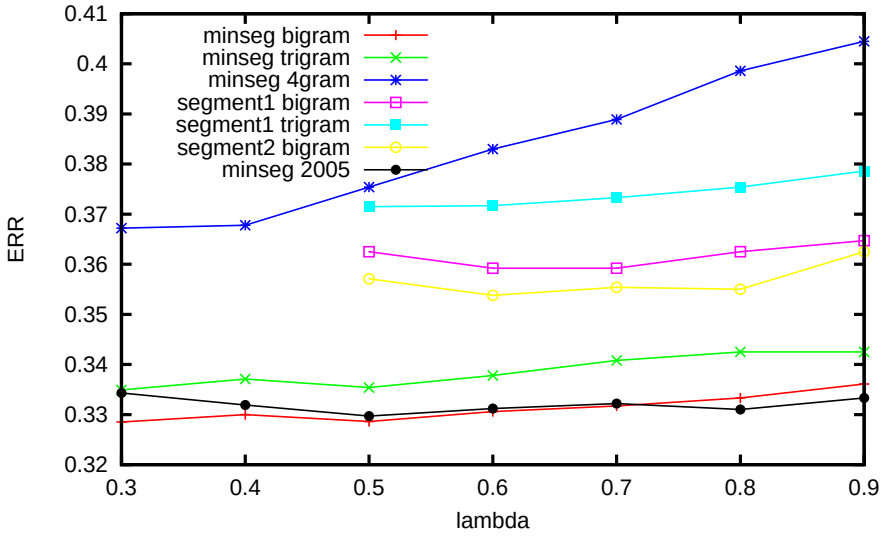
Další možnost jak ovlivnit segmentaci je chování algoritmu pro spojování segmentů po segmentu ticha. Za ním zpravidla následuje několik krátkých segmentů, ty by se podle původního algoritmu spojili se segmentem ticha a došlo by k posunu hranice segmentu. Je tedy možno tento algoritmus modifikovat, aby tyto segmenty nespojoval.

7.1.3 Výsledky

Z možností ladění segmentace a mediánového filtru jsem otestoval tyto systémy:

segment1 Minseg = 4, normální mediánový filtr, krátké segmenty po segmentu ticha se s tímto segmentem nespojují.

segment2 Minseg = 4, upravený mediánový filtr, krátké segmenty po segmentu ticha se s tímto segmentem spojují.



Obrázek 7.1: Srovnání výsledků systémů f_0, E

minseg Systém pracující s minimální délkou segmentu (minseg=1). Tento systém používá stejnou segmentaci i mediánový filtr jako segment2.

minseg 2005 Stejný systém jako minseg používající bigramový model, ovšem trénován a testován na NIST SRE 2005 datech.

Systém *minseg 2005* slouží pro porovnání vlivu použitých dat na výsledky.

V experimentech jsem se také pokoušel zjistit závislost výsledků na použití bigramového, trigramového nebo 4-gramového modelu.

Výsledky experimentů jsou vyneseny v grafu 7.1.3. Na ose x je parametr adaptace z UBM `lambda`. Na ose y je hodnota ERR (Equal eRror Rate).

Když porovnáme systémy lišící se segmentací *segment1 bigram* (fialový), *segment2 bigram* (žlutý) a *minseg bigram* (červený) je vidět postupný pokles chybovosti v závislosti na nastavení segmentace z $ERR = 35,92\%$ na $ERR = 32,85\%$.

Ze srovnání systémů založených na stejné segmentaci lišící se řádem použitého jazykového modelu například pro *minseg bigram* (červený), *minseg trigram* (zelený) a *minseg 4-gram* (modrý) jsou vidět, že nejlepší výsledky poskytuje bigramový model. Pokles ze $ERR = 36,72\%$ na $ERR = 32,85\%$.

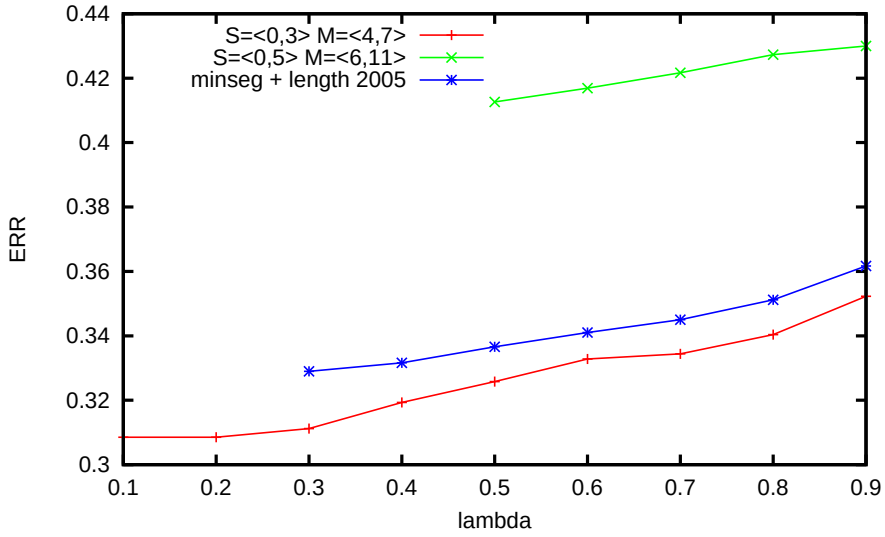
Průběh chybovosti pro systémy *minseg bigram* trénované a testované na NIST SRE 2005 a 2006 se liší pouze desetiny procenta.

7.2 Základní systém f_0, E + délka segmentu

Ke dvojici trend základního tónu a trend energie je přidána délka daného segmentu. Segmentace je stejná jako u systému minseg (kap. 7.1.3). Kvantizace délky segmentu na základě analýzy histogramu délek segmentu v trénovacích datech je popsána v kapitole 4.4.

V grafu 7.2 jsou srovnány tři systémy první (červený) používá kvantizační úrovně:

- krátký segment - délka 0 až 3 rámce
- střední segment - délka 4 až 7 rámce



Obrázek 7.2: Srovnání výsledků systémů f_0, E + délka segmentu

- dlouhý segment - delší než 8 rámců

vypočtené analýzou histogramu délek segmentů na trénovacích datech.

Druhý systém (zelený) používá odhadnuté hranice 6 a 12 rámců.

Třetí systém (modrý) je založen na NIST SRE 2005 datech a používá následující rozložení kvantizačních úrovní:

- krátký segment - délka 0 až 2 rámce
- střední segment - délka 3 až 5 rámce
- dlouhý segment - delší než 6 rámců

Na ose x je parametr adaptace z UBM `lambda`. Na ose y je hodnota ERR (Equal eRror Rate).

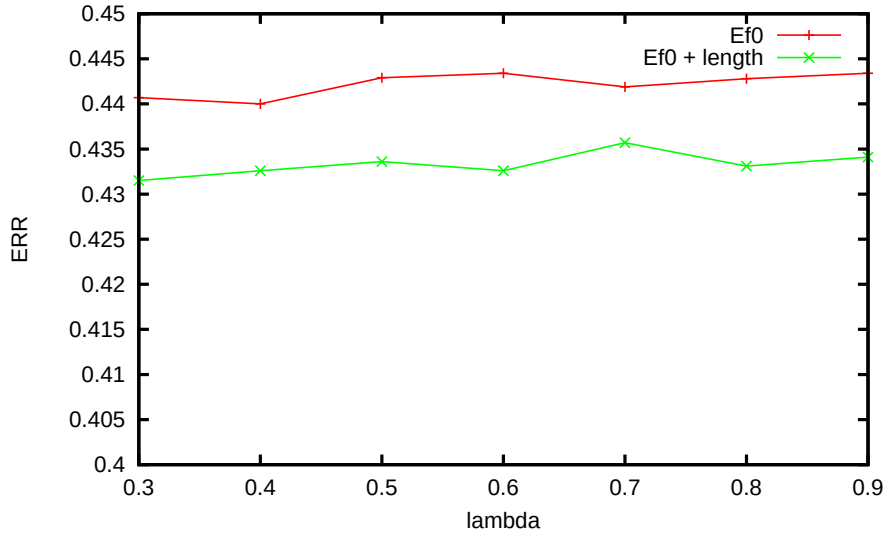
Ve srovnání se základním systémem není minimální chybovost kolem $lambda = 0,5$, ale leží v oblasti nejnižších hodnot parametru adaptace lambda. Pro základní systém s délkou segmentu je tedy nutné hodně vyhlazovat pomocí UBM.

Nejmenší chybovost systému založeného na odhadnutých hranicích byla $ERR = 41,26\%$. Pro hranice určené analýzou histogramu poklesla na $ERR = 30,85\%$ což je zlepšení o více než 10%.

Průběhy výsledků systému trénovaného a testovaného na datech z roku 2005 jsou asi o jedno procento horší než výsledky lepšího ze systémů trénovaného a testovaného na datech z roku 2006, ale křivka vykazuje stejný tvar.

7.3 Segmentace podle E

Jako alternativní postup k segmentaci podle průběhu základního tónu můžeme použít segmentaci podle průběhu energie. Ta pracuje stejně jako segmentace podle průběhu základního tónu, jen je místo průběhu základního tónu používán průběh energie.



Obrázek 7.3: Srovnání výsledků systémů založených na segmentaci podle průběhu energie

Oba testované systémy byly založeny na bigramovém jazykovém modelu a používaly stejné parametry segmentace jako systém *minseg*(kap. 7.1.3). První systém používal pouze příznaky ze segmentace.

Druhý systém k příznakům ze segmentace přidává délku segmentu. Analýzou histogramu délek segmentů trénovacích dat dostáváme kvantizaci:

- krátký segment - délka 0 až 4 rámce
- střední segment - délka 5 až 9 rámců
- dlouhý segment - delší než 10 rámců

Průběhy ERR těchto systémů v závislosti na nastavení adaptační konstanty `lambda` jsou zobrazené v grafu 7.3.

Když srovnáme systém minseg bigram s kapitoly 7.1.3 se systémem Ef0, dostáváme o více než 11% vyšší chybovost. Porovnání výsledků po přidání délky segmentu k základnímu systému vychází pro segmentaci podle základního tónu dokonce o více než 12% lépe. Z toho docházím k závěru, že segmentace podle energie není vhodná.

7.4 Shrnutí výsledků

Nejlepších výsledků dosahují bigramové modely, protože pro natrénování modelů vyšších řádů není dostatek dat.

Nejlepším základním systémem f_0, E je systém označený jako minseg. Úpravami v segmentaci a filtraci základního tónu bylo dosaženo zlepšení výsledků o tři procenta oproti systému segment1 a je pravděpodobné, že další práce by mohla přinést ještě lepší výsledky.

Přidáním délky segmentu dosáhneme zlepšení výsledků.o dvě procenta, ale je nutné přesně určit kvantizační hranice.

Při použití NIST SRE 2005 dat dosáhneme obdobných výsledků jako při použití NIST SRE 2006 dat.

Kapitola 8

Systémy založené na segmentaci fonémového rozpoznávače

8.1 Fonémový rozpoznávač

V experimentech byl použit výstup maďarského fonémového rozpoznávače [11] vyvinutého skupinou Speech@FIT [12].

Důvodem pro maďarský rozpoznávač je rozpoznání 61 fonémů, což je více než je běžné v ostatních jazycích. Pro rozpoznání řečníka není důležité znát jazyk jakým daný řečník mluví a proto si můžeme dovolit použít fonémový rozpoznávač natrénovaný na jiný jazyk.

Pro rozpoznání fonému se nepoužívá pouze příznakový vektor z jednoho rámce, ale i příznakové vektory z rámců, které daný rámec předchází a následují. Jak je vidět z obrázku 8.1 kontext je dělen na dvě poloviny (levou a pravou), kde každá je zpracovávána odděleně. Tomuto postupu se říká *split temporal context* (STC) (další informace jsou v [8]). Postupem STC se snižuje množství dat potřebných k trénování neuronových sítí a výpočetní náročnost.

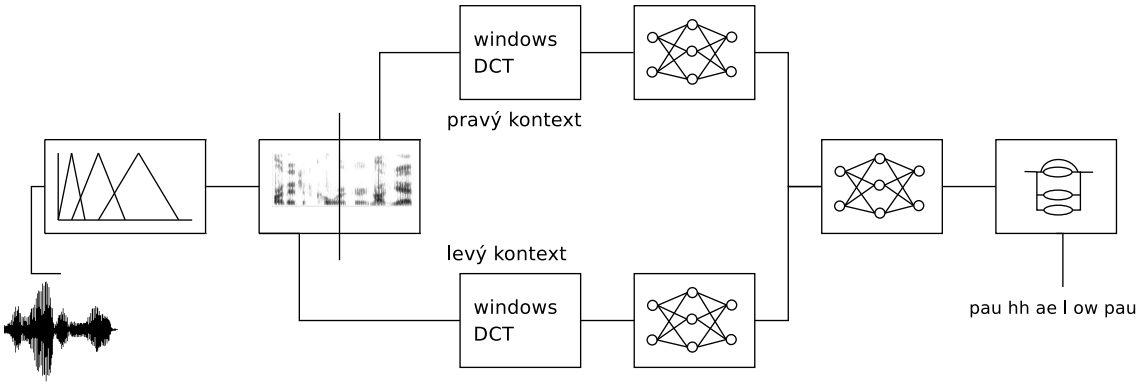
Vstupní řečový signál je filtrován melovskou bankou filtrů. Pro oddělení je použito Hanningovo okno. Poté je zredukována dimenze příznakového vektoru pomoci diskrétní kosinové transformace. Následuje normalizace hodnot ve dvou krocích implementovaná pomocí tří neuronových sítí. Nejprve je zvlášť normalizován levý a pravý kontext a v druhém kroku je výstup spojen pomocí třetí neuronové sítě. Všechny sítě jsou třívrstvé a obsahují 1500 neuronů. Výstup je generován pomocí HMM dekodéru obsahující tři stavy pro každý model.

Pro použití fonémového rozpoznávače v rozpoznání mluvčího je nutno nastavit konstantu vyjadřující jak často se může generovat další foném na jinou hodnotu než při použití k jiným účelům (rozpoznání řeči, identifikace jazyka atd.).

8.2 Základní systém

Kromě segmentace podle základního tónu je možno využít výstup z fonémového rozpoznávače. Ten obsahuje pro každý segment trojici začátek segmentu, konec segmentu a rozpoznaný foném.

Segmentace probíhá ve třech krocích. Nejprve je načten jeden segment z výstupu fonémového rozpoznávače, poté je ze souboru z dvojicemi základní tón, energie vypočítán trend daného segmentu a nakonec je vygenerován výstupní příznakový vektor obsahující v případě základního systému trend základního tónu a trend energie.



Obrázek 8.1: Schéma fonémového rozpoznávače

8.3 Rozšíření základního systému

K základní segmentaci můžeme přidat

- délku segmentu
- foném
- délku segmentu a foném

Délka fonému je stejně jako u segmentace podle f_0 kvantována do tří hladin. Kvantizační úrovně určíme pomocí histogramu délek segmentů trénovacích dat tak, aby do každé úrovně připadla třetina segmentů. Zde vyšlo rozdělení

- krátký segment - délka 0 až 4 rámce
- střední segment - délka 5 až 7 rámců
- dlouhý segment - délka 8 a více rámců

8.4 Výsledky

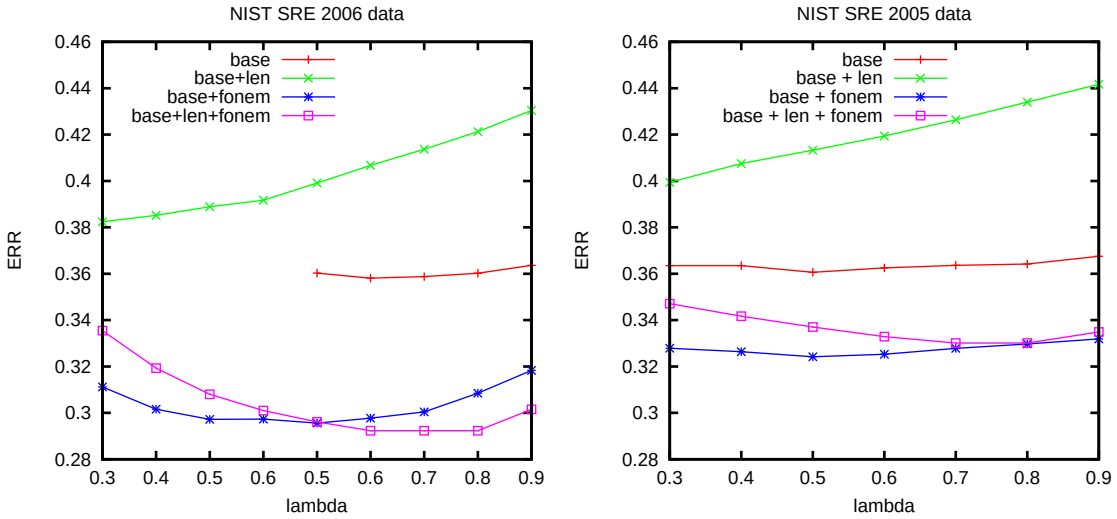
Porovnání chybovosti ERR (Equal eRror Rate) systémů založených na segmentaci podle fonémového rozpoznávače je v grafu 8.2. Na ose x je parametr adaptace z UBM `lambda` a na ose y je hodnota ERR (Equal eRror Rate).

U všech systémů je použit bigramový jazykový model, protože vyšší řád modelu nepřináší zlepšení výsledků rozpoznání. Základní systém založený na trigramovém modelu má chybovost $ERR = 38,44\%$ a ten samý systém založený na bigramovém modelu má chybovost pouze $ERR = 35,81\%$. Na obrázku 8.2 na trigramech založený systém není.

Nejlepších výsledků dosahuje systém používající trend základního tónu, trend energie, délku segmentu a foném, který je jen nepatrně lepší než systém používající trend základního tónu, trend energie a foném.

Základní systém používající jen trend základního tónu a trend energie je lepší pokud použijeme segmentaci podle základního tónu. Přidáním fonému dojde k zásadnímu zlepšení výsledků.

Systémy je nutno testovat v celém rozsahu adaptační konstanty lambda, protože každý systém potřebuje jiné nastavení.



Obrázek 8.2: Srovnání výsledků systémů založených na segmentaci podle fonémového rozpoznávače

Při použití NIST SRE 2005 dat dosáhneme obdobných průběhů výsledků jako při použití NIST SRE 2006 dat, ovšem tyto průběhy jsou posunuty směrem k horším výsledkům. To je velmi dobře vidět z obrázku 8.2.

Kapitola 9

Spojení se SpkID založeném na akustickém modelování

Jako zástupce systému založeného na akustických příznacích byl vybrán systém označovaný jako BUT-GMM. vyvinutý skupinou Speech@FIT [12] pro NIST SRE 2006 evaluaci.

9.1 BUT-GMM

Systém využívá GMM pro tvorbu modelu mluvčího a modelu UBM. Důležitým prvkem systému je adaptace modelů mluvčích z UBM.

Podle [3] je hlavním problémem při rozpoznání mluvčího odstranit vlivy prostředí, ve kterém probíhalo pořizování nahrávek. Toho se dosahuje pomocí

- cepstral mean substraction
- feature warpingu
- RASTA filtrace (RelAtive SpecTrAl)
- HLDA (Heteroscedastic Linear Discriminant Analysis)
- feature mappingu
- eigenchannel adaptation

Podrobný popis těchto technik je v [3].

Jako příznaky jsou použity Mel-frekvenční kepstrální koeficienty (MFCC).

9.2 Výsledky

Pro spojení se systémem BUT-GMM byly vybrány dva systémy. První označený jako *f0E-len* je systém založený na segmentaci podle základního tónu s délkou segmentu (nejlepší systém popsaný v kapitole 7.2). Druhý systém označený jako *fonem* je systém založený na výstupu fonémového rozpoznávače s délkou segmentu a fonémem (systém base-len-fonem z kapitoly 8.4).

Experiment probíhal na NIST SRE 2006 datech (kap. 6.1.1).

Jak je vidět v tabulce 9.2 spojením systému založeného na prosodických příznacích s tradičním systémem pro rozpoznání řečníka dojde ke zlepšení úspěšnosti rozpoznání. A to

Systém	**ERR**	**DCF**
BUT-GMM	3,45 %	1,78 %
BUT-GMM + f0E-len	3,45 %	1,78 %
BUT-GMM + fonem	3,24 %	1,77 %

Tabulka 9.1: Výsledky spojení s tradičním systémem SpkID

při spojení se systémem fonem ze ERR = 3,45 % na ERR = 3,24 %, což je relativní zlepšení o 6,1 %.

Kapitola 10

Závěr

10.1 Shrnutí výsledků experimentů

V tabulce 10.1 jsou porovnány výsledky jednotlivých systémů na základě použité segmentace. Nejlepších výsledků dosahuje segmentace podle základního tónu a nejhorších segmentace na základě energie.

V tabulce 10.2 jsou jednotlivé systémy porovnány v závislosti na jakých datech jsou trénovány a testovány. Výsledky jednotlivých systémů jsou srovnatelné. Zajímavým bodem u výsledků získaných na NIST SRE 2005 datech je lepší úspěšnost systému založeného na segmentaci fonémového rozpoznávače a fonému než systému založeného na segmentaci fonémového rozpoznávače, délce segmentu a fonému.

Po spojení nejlepšího systému založeného na segmentaci z fonémového rozpoznávače se systémem založeným na akustických příznacích BUT-GMM se zlepší uspěšnost rozpoznání mluvčího o více než 6 % (viz. kap. 9).

10.2 Pokračování projektu

Výzkumná skupina Speech@FIT [12] se pravidelně zúčastňuje evaluací systémů pro rozpoznání řečníka pořádaných organizací NIST. V systému skupiny Speech@FIT nebyly v minulosti použity prosodické příznaky a já doufám, že v příštích letech již budou tyto příznaky obsaženy. Dobrým důvodem pro jejích použití je relativní zlepšením výsledků o 6 % (viz. kap.9).

V kapitole 7.1 je vidět, že výsledky systému značně ovlivňují modifikace provedené v segmentaci. Z toho usuzuji, že segmentace podle základního tónu je otevřený problém a hledáním dalších modifikací je možno dosáhnout zlepšení výsledků.

V [4] je popsán ještě jeden postup tvorby prosodických příznaků založený na Dynamic

Systém	**Použitá segmentace**					
	podle f_0		**podle E**		**podle fonem.**	
	ERR	DCF	ERR	DCF	ERR	DCF
základní segmentace	32,85%	61,62%	44,00%	83,54%	35,81%	67,39%
délka segmentu	30,85%	57,56%	43,15%	81,53%	38,24%	72,17%

Tabulka 10.1: Srovnání výsledků segmentačních metod

Systém	NIST SRE data			
	2005		**2006**	
	ERR	DCF	ERR	DCF
základní f0E	32,97%	62,09%	32,85%	61,62%
základní f0E + délka	32,90%	62,21%	30,85%	57,56%
základní fonem	36,06%	67,94%	35,81%	67,39%
základní fonem + délka	39,94%	75,28%	38,24%	72,17%
základní fonem + fonem	32,42%	61,20%	29,56%	55,89%
základní fonem + délka a fonem	33,01%	62,47%	29,23%	55,09%

Tabulka 10.2: Srovnání výsledků v závislosti na použitých datech

Time Warping. Tímto postupem jsem se v práci nezabýval, a proto by bylo vhodné se v dalších navazujících projektech na tento postup zaměřit.

Podle [4] je úspěšnost metod založených na prosodických příznacích silně závislá na množství trénovacích dat. Další zlepšení úspěšnosti rozpoznávání by tedy mělo přinést natrénování modelů na větším množství promluv od daného mluvčího.

Dodatek A

Algoritmus detekce základního tónu v praatu

Podrobné hodnocení algoritmu a jeho srovnání s ostatními metodami je v [1].

A.1 Parametry

TimeStep perioda detekce základního tónu ze vstupního signálu.

MaximumNumberOfCandidatesPerFrame maximální počet kandidátů na rámec.

MinimumPitch určuje minimální hodnotu detekovaného základního tónu.

MaximumPitch určuje maximální hodnotu detekovaného základního tónu.

VoicingThreshold práh pro znělý rámec.

SilenceThreshold práh pro neznělý rámec.

OctaveCost favorizuje vyšší hodnoty základního tónu.

VoicedUnvoicedCost cena přechodu ze znělého na neznělý rámec.

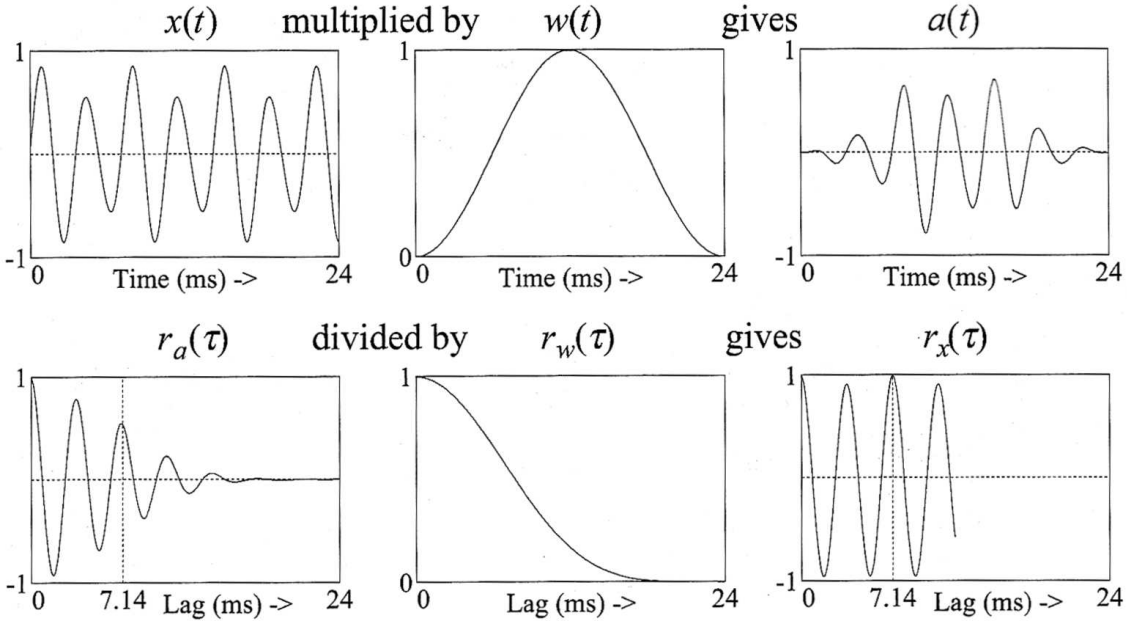
OctaveJumpCost cena přechodu na vyšší hodnoty základního tónu.

A.2 Průběh algoritmu

1. Předzpracování - Hannigovým oknem se odstraní hodnoty signálu od 95% do 100% Nyquistovy frekvence. To probíhá v kmitočtové oblasti.

2. Výpočet maximální absolutní hodnoty (použito v detekci zda je rámec znělý nebo neznělý).

3. Dělení signálu na rámce. Jeden rámec každých *TimeStep* sekund. Pro každý rámec se počítá nejvýše *MaximumNumberOfCandidatesPerFrame* párů zpoždění, které jsou kandidáty pro hodnoty periody základního tónu. Odstranění neznělých kandidátů probíhá později. Pro každý rámec:



Obrázek A.1: Na signál $x(t)$ je aplikováno okno $w(t)$. Tím dostaneme signál $a(t)$. Výsledný průběh $r_x(\tau)$ dostaneme dělením autokorelace $r_a(\tau)$ signálu $a(t)$, autokorelací okna $r_w(\tau)$.

(a) Vem segment signálu, kde délka segmentu je omezena parametrem MinimumPitch. Okno by mělo být dlouhé tak, aby se do něj vlezly tři periody *MinimumPitch*.

(b) Odečti střední hodnotu.

(c) Je dokázáno, že první kandidát je vždy neznělý. Váha kandidáta je určena pomocí dvou parametrů *VoicingThreshold* a *SilenceThreshold*.

(d) Vynásob oknem:

$$a(t) = (x(t_{mid} - \frac{1}{2}T + t) - \mu_x)w(t) \tag{A.1}$$

kde $x(t)$ je vstupní signál, T je délka rámce, t_{mid} je střed rámce, μ_x je střední hodnota rámce a $w(t)$ je okénková funkce

$$w(t) = \frac{1}{2} - \frac{1}{2}\cos\frac{2\pi t}{T} \tag{A.2}$$

(e) Naplň polovinu délky okna nulami.

(f) Doplň segment nulami tak, aby jeho délka byla mocninou dvou.

(g) Proveď rychlou Fourierovu transformaci.

(h) Umocni vzorky na druhou ve frekvenční oblasti.

(i) Proveď rychlou inverzní Fourierovu transformaci. Dostáváme autokorelaci $r_a(\tau)$ signálu.

(j) Dělíme autokorelaci signálu $r_a(\tau)$ autokorelací okna $r_w(\tau)$. Dostáváme vzorkovanou hodnotu $r_x(\tau)$ (viz. obrázek A.1)

(k) Najdi hodnoty a umístění maxim v nevzorkované průběhu $r_x(\tau)$. Hledej pouze umístění maxim, které odpovídá základnímu tónu mezi *MinimumPitch* a *Maxi-*

mumPitch. Pro neznělé je síla definována jako:

$$R \equiv VoicingThreshold + \max\left(0,2 - \frac{\frac{\text{local absolute peak}}{\text{global absolute peak}}}{\frac{SilenceThreshold}{1+VoicingThreshold}}\right) \tag{A.3}$$

a pro znělé

$$R \equiv r(t_{max}) - OctaveCost^2 \cdot \log(MinimumPitch \cdot \tau_{max}) \tag{A.4}$$

Po provedení kroku 3 pro každý rámec dvojice frekvence, síla (F_{ni}, R_{ni}), kde index n nabývá hodnot od 1 do počtu rámců a index i od 1 do počtu kandidátů na rámec. Je tedy z těchto kandidátů potřeba vybrat ten nejlepší ve vztahu k ostatním.

4. Pro každý rámec n, je p_n číslo mezi 1 a počtem kandidátů na rámec. Hodnoty $\{p_n | 1 \leq n \leq \text{počet rámců}\}$ definuje cestu přes kandidáty $\{(F_{np_n}, R_{np_n}) | 1 \leq n \leq \text{počet rámců}\}$. S každou možnou cestou je vázána cena

$$cost(\{p_n\}) = \sum_{n=2}^{numberOfFrames} transitionCost(F_{n-1,p_{n-1}}, P_{np_n}) - \sum_{n=1}^{numberOfFrames} R_{np_n} \tag{A.5}$$

kde funkce *transitionCost* je definována jako

$$transitionCost(F_1, F_2) = \begin{cases} 0 & \text{if } F_1 = 0 \text{ and } F_2 = 0 \\ VoicedUnvoicedCost & \text{if } F_1 = 0 \text{ xor } F_2 = 0 \\ OctaveJumpCost|^2 \log \frac{F_1}{F_2}| & \text{if } F_1 \neq 0 \text{ and } F_2 \neq 0 \end{cases} \tag{A.6}$$

Nejlepší cesta je taková, která má nejnižší cenu. Nejlepší cesta se dá najít pomocí postupů dynamického programování, jako je například pomocí Viterbiho algoritmus popsaný například v [7, str. 209].

Literatura

[1] Paul Boersma. Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. *Proceedings of the Institute of Phonetic Sciences of the University of Amsterdam*, (vol. 17), 1993.

[2] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[3] Lukáš Burget, Pavel Matějka, Petr Schwarz, Ondřej Glembek, and Jan Černocký. Analysis of feature extraction and channel compensation in gmm speaker recognition system. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7):1979–1986, 2007.

[4] Andre G. Adami; Radu Mihaescu; Douglas A. Reynolds; John J. Godfrey. Modeling prosodic dynamics for speaker recognition. *Acoustics, Speech, and Signal Processing*, 4:IV – 788–91, 2003.

[5] J. Gonzalez-Rodriguez, D. Ramos-Castro, D. Torre-Toledano, A. Montero-Asenjo, J. Gonzalez-Dominguez, I. Lopez-Moreno, J. Fierrez-Aguilar, D. Garcia-Romero, and J. Ortega-Garcia. On the use of high-level information for speaker recognition: the atvs-uam system at NIST SRE 2005. *IEEE Aerospace and Electronic Systems Magazine*, 22(1):15 – 21, January 2007.

[6] Alvin Martin, George Doddington, Terri Kamm, Mark Ordowski, and Mark Przybocki. The DET curve in assessment of detection task performance. pages 1895–1898, 1997.

[7] Josef Psutka; Luděk Muller; Jindřich Matoušek; Vlasta Radová. *Mluvíme s počítačem česky.* Academia, 2006.

[8] Petr Schwarz, Pavel Matějka, and Jan Černocký. Towards lower error rates in phoneme recognition. page 8, 2004.

[9] Andreas Stockle. Srilm - an extensible language modeling toolkit. *in Proc. Intl. Conf. Spoken Language Processing*, (September), 2002.

[10] WWW stránky. Artificial neural network from wikipedia, the free encyclopedia. `http://en.wikipedia.org/wiki/Artificial_neural_network`.

[11] WWW stránky. Phoneme recognizer based on long temporal context. `http://speech.fit.vutbr.cz/en/software/phoneme-recognizer-based-long-temporal-context`.

[12] WWW stránky. Speech@FIT. http://speech.fit.vutbr.cz/.

[13] WWW stránky organizace NIST. Speaker recognition evaluation. http://www.nist.gov/speech/tests/sre.